(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 449 349 A1**

# (12) EUROPEAN PATENT APPLICATION

(21) Application number: 91200523.8

(22) Date of filing: 12.03.91

(51) Int. Cl.$^5$: G09C 1/00, G09C 5/00,
H04L 9/30

(30) Priority: 15.03.90 NL 9000593

(43) Date of publication of application:
02.10.91 Bulletin 91/40

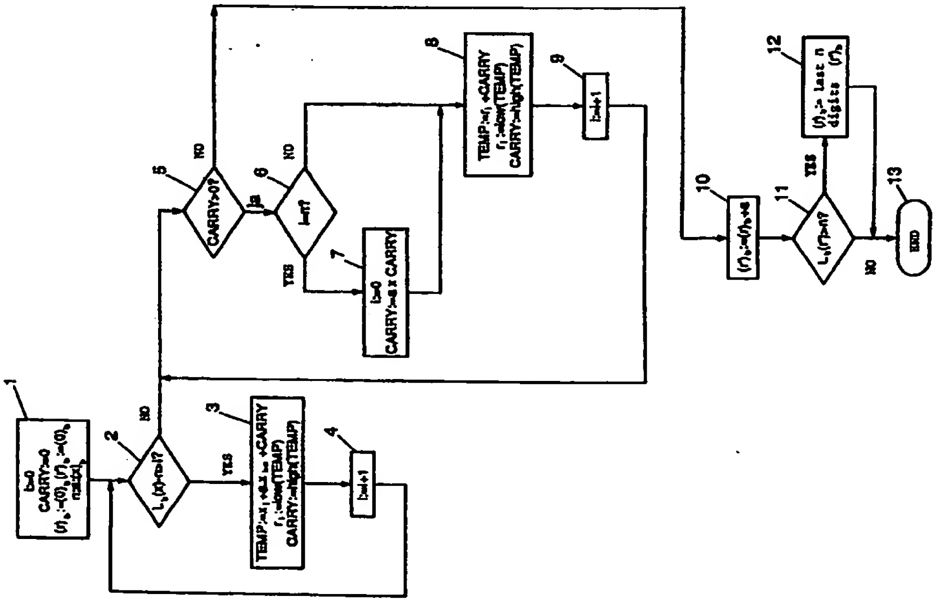
(84) Designated Contracting States:
AT BE CH DE DK ES FR GB GR IT LI LU NL SE

(71) Applicant: Koninklijke PTT Nederland N.V.
P.O. Box 95321
NL-2509 CH The Hague(NL)

(72) Inventor: The inventor has agreed to waive
his entitlement to designation

## (54) Method for the modular reduction of numbers.

(57) In cryptographic techniques which are based on the discrete logarithm problem, use is made of exponentiation modulo large number. If, in this method, the exponentiation is first carried out completely and the modular reduction is only carried out thereafter, this requires very considerable computation time and a very large memory capacity. It is known that the exponentiation can be accelerated by iteratively multiplying and squaring, with a modular reduction after each step. The invention provides a method of also accelerating the modular reduction at the same time, as a result of which the exponentiation modulo large number can be further accelerated. For this purpose, the invention describes a protocol for a modular reduction of a 2n-digit number x in a number system with base b to obtain an n-digit remainder, in which use is made of a specific modulus p which satisfies p = $b^n$ - a, where 0 < a < b. The method according to the invention does not reduce the security of the cryptographic system for which the modular exponentiation is carried out.



EP 0 449 349 A1

Rank Xerox (UK) Business Services

The invention relates to a method for the modular reduction of a not more than 2n-digit number x to obtain a not more than n-digit remainder in accordance with the formula
$r \equiv x \pmod p$, where p is the chosen modulus, for the purpose of cryptographic calculations which are based on the general discrete logarithm problem in a number system with base b.

In cryptographic techniques, use may be made of exponentiation modulo large prime number. An example of this is a cryptographic technique which is based on the discrete logarithm problem which is described by w. Diffie and M.E. Hellman in the paper entitled "New directions in Cryptography" in IEEE Trans. on Information Theory, vol. IT-22, pages 644-654, 1976.

This paper describes a discrete exponentiation modulo prime number p for use in a public key distribution system. This so-called DH system makes use of publicly known messages in the following way in order to construct a secret common key. Choose a prime number p for which it is true that
p - 1 has at least one large prime factor, and an integer z from the set {2,3,......,p - 1}. These two numbers are made public. Every user now arbitrarily chooses a number x from the set {2,3,.......,p - 2}, keeps it secret and calculates the number $y = z^x \pmod p$. Two users a and b send their numbers y, i.e. $y_a$ and $y_b$, respectively, to each other. With the aid of his own value of x, $x_a$, and the $y_b$ received, user a can calculate a secret key $k_a$ in accordance with
$k_a = {y_b}^{x_a} \equiv z^{x_a x_b} \pmod p$.

With the aid of his own value of x, $x_b$, and the $y_a$ received, user b can calculate a secret key inaccordance with: $k_b = {y_a}^{x_b} \equiv z^{x_b x_a} \pmod p$. It will be clear that $k_a = k_b = k$ and that the users have constructed a common secret key in this way.

Because the taking of the discrete logarithm has hitherto been regarded as virtually impossible, it is also virtually impossible to calculate the integers $x_a$ and $x_b$ or the key k starting from the numbers $y_a$ and $y_b$. Methods other than that of taking the discrete logarithm to calculate these values have hitherto not been known. Exponentiation modulo large prime number, the exponentiation being carried out first and the result thereby obtained then being reduced results quite quickly in unacceptably long computation times, while the memory capacity required in that case becomes very large.

In "The Art of Computer Programming", Vol. 2: Seminumerical Algorithms, 2nd edition, Addison Wesley, 1981, D.E. Knuth describes how an exponentiation can be imagined as built up of repeated multiplication and squaring. He also describes how modular exponentiation can be simplified by making use of modular reduction in every multiplication step and squaring step. Said modular reduction generally takes place by making use of a division algorithm. If the modulus p is the divisor and the number to be reduced is x, the remainder obtained after division gives the desired results; or:

$$x = q.p + r,$$

where q is the quotient, this being written in the present description as

$$r \equiv x(\text{mod } p).$$

If the opportunity is seen of accelerating the modular reduction, the exponentiation can also be accelerated and the object of the invention is therefore to provide a method for the rapid modular reduction of a large number from a number system with base b by means of an iterative division algorithm. Such division algorithms are known per se from, for example, the paper entitled "Fast algorithms for implementing RSA public key cryptosystem" by S.B. Mohan and B.S. Adiga in Electronics Letters, vol. 21, no. 17, August 1985, which describes an iterative division algorithm for use in the RSA system, which algorithm has been developed exclusively for the binary system and for a composite modulus which is made up of the product of two large prime numbers and a number of smaller prime numbers. The paper entitled "A practical fast exponentiation algorithm for public key" by H.R. Chivers, International Conference on Secure Communications Systems, London, 22-23 February 1984, furthermore describes an accelerated division algorithm for use in encoding system of Diffie and Hellman. In this case, however, it is proposed to make use of remainder tables, and this is often undesirable in view of the memory space required for such tables. The object of the invention is, more particularly, to provide a method for fast modular reduction which is efficient to implement in software, preferably with a minimum of program lines, which is of great importance, in particular, if said program is present in the processor present on a so-called smart card. This object must at the same time be fulfilled without diminishing the safety of the cryptographic protocol, for the purpose of which the modular reduction is carried out.

For a general description of the way in which a cryptographic system can be used in combination with a smart card, reference is made to the paper entitled "The Smart Card: A high security tool in EDP" by R.C. Ferreira in Philips Telecommunication and Data Review, PTR, Volume 47, no. 3, September 1989, pages 1 - 19. However, said paper describes the use in cryptographic systems in which the discrete logarithm problem does not occur.

In order to fulfil the objects described, the invention provides a method of the abovementioned type in which it holds true for the modulus p that $p = b^n - a$, where $0 < a < b$. More particularly, the invention is characterised in that the 2n-digit number x is always split into two n-digit numbers $x_H$ and $x_L$ in accordance with the formula

$$x = (x_H \cdot b^n) + x_L,$$

where

$$x_H = \sum_{i=0}^{n-1} x_{i+n} \cdot b^i \text{ and } x_L = \sum_{i=0}^{n-1} x_i \cdot b^i \text{ where } 0 \leq x_i,\ x_{i+n} < b;$$

in that the remainder r is determined by first calculating:

$$r_{temp} = x_L + a \cdot x_H;$$

in that it is then determined whether the following is fulfilled:

$$r_{temp} < b^n;$$

in that, if this is the case, it holds true that:

$$r = r_{temp};$$

in that, if this is not the case, an integer c is calculated which is determined by the value of $r_{temp}/b^n$ rounded downwards to the nearest integer and $r_{temp}$ is calculated in accordance with:

$$r_{temp} = r_{temp} - c \cdot b^n$$

and in that this step is repeated until the following is fulfilled:

$$r_{temp} < b^n,$$

whereafter it holds true that:

$$r = r_{temp}.$$

The invention can be explained in greater detail in the following manner.

Suppose that x, the number to be reduced, and r, the remainder, are positive integers and that their representation in the system with base b is given by:

$$(x)_b = \sum_{i=0}^{2n-1} x_i \cdot b^i \text{ and } (r)_b = \sum_{i=0}^{n-1} r_i \cdot b^i \text{ where } 0 \leq x_i,\ r_i < b.$$

For the modulus p it holds true that

$$(p)_b = \sum_{i=0}^{n-1} p_i \cdot b^i = b^n - a \text{ where } 0 < a < b \text{ and } n > 1.$$

The modular reduction can now be carried out as follows:
$(r)_b \equiv (x)_b \pmod{(p)_b} = (x)_b - ((k)_b {}^\bullet (P)_b) =$
$(X)_b - (k)_b {}^\bullet b^n + (k)_b {}^\bullet a$, where $(k)_b$ is so chosen that this equation is fulfilled.

If it is assumed that use is made of registers having a length of n digits, the number $(x)_b$ can be reproduced by two n-digit registers $(x_H)_b$ and $(x_L)_b$ as:

$(x)_b = ((x_H)_b \,|\, 1(x_L)_b = (x_H)_b {}^\bullet b^n + (x_L)_b$,

with

$$(x_H)_b = \sum_{i=0}^{n-1} x_{i+n} \cdot b^i \text{ and } (x_L)_b = \sum_{i=0}^{n-1} x_i \cdot b^i$$

From this it follows that if $(x_H)_b = (k)_b$ is chosen, the reduced $(x)_b$ can be obtained by adding the number $a^\bullet\ (x_H)_b$ to $(x_L)_b$ and by continuing to do this with the $(x_H)_b$ produced after the addition until $(x_H)_b = 0$ and therefore $(x)_b < b^n$.

The flow diagram shown in the drawing reproduces the method according to the invention in more detail for a number in the system with base b.

If it is true for a number w in the system with base b that: $w = y^\bullet b + x$, the functions low ( ) and high ( ) are defined as:

low(w) = x and high(w) = y.

The length $n = L_b(s)$ of a number $(s)_b$ in the system with base b follows from:

$L_b(s) = \max\{i|(s)_b \geq b^i\} + 1$.

The modular reduction $(r)_b \equiv (x)_b \pmod{(b^n - a)_b}$ then proceeds in the way shown in the flow diagram.

Block 1 shows that, at the beginning of the modular reduction of a number $(x)_b$, the loop variable i and the variable CARRY are equal to 0 and the remainder values $(r)_b$ and $(r')_b$ are equal to $(0)_b$, n and a being predetermined constants which determine the modulus. For $(x)_b$ and for $(r)_b$, the formulae already given above apply:

$$(x)_b = \sum_{i=0}^{2n-1} x_i \cdot b^i \text{ and } (r)_b = \sum_{i=0}^{n-1} r_i \cdot b^i, \text{ where } 0 \leq x_i,\ r_i < b.$$

In block 2, whether the length of $(x)_b$, i.e. $L_b(x)$, minus n is greater than i is determined. If that is the case, a temporary number TEMP is determined in block 3 in accordance with $\text{TEMP} = x_i + a^\bullet x_{i+n} + \text{CARRY}$, $r_i$ then becoming equal to low(TEMP) and the variable CARRY equal to high(TEMP).
Then the loop variable i in block 4 is increased by 1 so that it becomes equal to i + 1, and block 2 is reverted to. When, at a certain instant, it is the case in block 2 that $L_b(x) - n \leq i$, the modular reduction is continued in block 5 by determining whether CARRY is greater than 0, and if this is not the case, the remainder $(r)_b$ is known in principle. If CARRY is in fact greater than 0, a determination is first made in block 6 of whether i is equal to n. If this is the case, the loop variable i is again put equal to 0 in block 7 and the

variable CARRY is multiplied by the constant a, after which block 8 is proceeded to. If it is found in block 6 that i is not equal to n, block 8 is also proceeded to. In block 8, TEMP is put equal to
TEMP = $r_i$ + CARRY, $r_i$ then becoming equal to low(TEMP) and the variable CARRY becomes equal to high(TEMP). Then the loop variable i is increased by 1 in block 9 so that it becomes equal to i + 1, and the loop is run through again, starting at block 5 until it is finally true that CARRY $\leq$ 0, which in principle determines the remainder. Once the remainder $(r)_b$ has been determined in block 5, there is the possibility that the remainder $(r)_b$ is nevertheless greater than the modulus p, the chance of this being $a/b^n$.

For this reason an auxiliary remainder $(r')_b = (r)_b + a$ is calculated in block 10 and in block 11 a check is made on whether the length of $(r')_b$, i.e. $L_b(r')$, is in fact greater than n and if this is not the case, the calculated remainder $(r)_b$ is already the true remainder. However, if $L_b(r')$ is in fact greater than n, the true remainder $(r)_b$ is calculated in block 12 by taking the last n digits $(r')_b$. Block 13 indicates the end of the calculation of the remainder $r_b$. As will be explained further below, the steps according to the blocks 10, 11 and 12 are in many cases superfluous because if it is found in block 5 that the CARRY is not greater than 0, an n-digit remainder has already been calculated, and this is in principle the object of the method according to the invention.

With the method according to the invention as explained above, a modular reduction is possible with the aid of a limited number of multiplications if a > 1 and even no multiplication at all if a = 1. If a = 1, the loop formed in the flow diagram formed by the blocks 2, 3 and 4 is executed not more than n times, without multiplications having to be carried out and block 7 is executed not more than once, also without multiplications being necessary. If a > 1, the loop formed by the blocks 2, 3 and 4 in the flow diagram is executed not more than n times, not more than n multiplications having to be carried out in block 3 and block 7 is executed not more than twice, not more than 2 multiplications being necessary, so that if a > 1, the maximum number of multiplications needed is n + 2.

The flow diagram described above provides an explanation of the method according to the invention for a calculation in software, the entire calculation of the n-digit remainder $(r)_b$ of the 2n-digit number $(x)_b$ in the system with base b being carried out digit by digit. However, it will be immediately obvious to those skilled in the art that the method according to the invention can also be implemented in an extremely efficient way in hardware, in which case multiplications by n-digit numbers are carried out directly. The invention will be illustrated further with reference to two numerical examples in the decimal system, use being made of such multiplications by n-digit numbers for the sake of simplicity.

In the decimal system, b = 10. Furthermore, the number 9991 is chosen for the modulus $(p)_{10}$, so that n = 4 and $a = b^n - (p)_{10} = 10^4 - 9991 = 0009$.

In the first example, the remainder of the number 99980001 is sought.

According to the first step of the algorithm it is calculated that:

$$(x_H)_{10} \cdot a + (x_L)_{10} = 9998 \times 0009 + 0001 = 00089983.$$

The carry, the newly-formed $(x_H)_{10}$, is thus found to be equal to 0008 and $(x_L)_{10}$ = 9983, and because the carry is greater than zero, the following calculation is carried out:

$$\text{carry} \times a + (x_L)_{10} = 0008 \times 0009 + 9983 = 00010055.$$

Again the carry, the newly-formed $(x_H)_{10}$ is greater than zero so that a calculation is again carried out:

$$\text{carry} \times a + (x_L)_{10} = 0001 \times 0009 + 55 = 0064.$$

The carry is now equal to zero and the remainder is therefore known in principle and is equal to the last $(x_L)_{10}$ calculated = 0064. As has already been noted above in the description of the flow diagram, there is a slight chance that the calculated remainder $(r)_{10}$ is greater than $(p)_{10}$. However, because the invention is in principle aimed at the fast modular reduction of a 2n-digit number to an n-digit number, it is of less importance whether the n-digit remainder found is also the true remainder or whether it yet again contains more than the modulus. In the first example given above, it is clear that $(r)_{10} < (p)_{10}$, so that an explanation of the steps according to the blocks 10 - 12 in the flow diagram is of little interest for this example. However, by adding the value of a to the n-digit remainder found and determining whether the length of the auxiliary remainder $(r')_{10}$ thereby calculated is greater than n, the true remainder can always be determined, as will be illustrated by reference to the following example.

In the second example, the remainder of 19987 is sought. According to the first step of the algorithm, the following calculation is carried out:

$(x_H)_{10} \cdot a + (x_L)_{10} = 0001 \times 0009 + 9987 = 9996.$

The carry is now equal to zero and the remainder $(r)_{10}$ is therefore known in principle, but it is not known, however, if this is the true remainder. For this reason, the following calculation is furthermore first carried out:

$(r')_{10} = (r)_{10} + a = 9996 + 0009 = 10005.$

The value of $(r')_{10}$ is found to be greater than n and the true remainder $(r)_{10}$ is therefore determined by taking the last 4 digits of $(r')_{10}$, so that $(r)_{10}$ is found to be = 0005.

The modular reduction described above is of importance in order to be able to carry out cryptographic methods efficiently and rapidly, with use being made of exponentiation modulo large prime number such as the DH system described above. A number of other types of these cryptographic methods will be briefly discussed below.

A secret m can be exchanged by the so-called Three-Pass protocol with the aid of discrete exponentiation modulo known prime number p, possibly made public. This is done in the following way: As in the DH system, the users a and b both choose a secret arbitrary number x, $x_a$ and $x_b$ respectively, but they now both calculate a secret $x^{-1}$ which satisfies the relationship:

$x \cdot x^{-1} \equiv 1 \pmod{p - 1}.$

If user a wishes to send the secret message m to user b, he calculates

$y \equiv m^{Xa} \pmod{p}$

and sends it to b. User b now calculates:

$Z = y^{Xb} \equiv m^{Xa\ Xb} \pmod{p}$

and sends it back to a. User a now calculates:

$$z^{Xa^{-1}} = m^{Xa \cdot Xb \cdot Xa^{-1}} \equiv m^{Xb} \pmod{p}$$

and sends it back to b. From this, user b can calculate the secret message via:

$$(m^{Xb})^{Xb^{-1}} = m^{Xb \cdot Xb^{-1}} \equiv m \pmod{p}.$$

A third cryptographic system is known under the name of the Pohlig-Hellman (PH) system, and here again, use is made of a prime number p for which it holds true that p - 1 has at least one large prime factor. In the PH system, the users a and b choose secret numbers $x_a$ and $x_b$ so that it holds true that

$x_a x_b \equiv 1 \pmod{p - 1}.$

A secret message m can now therefore be exchanged between a and b with the aid of the following two functions:

The encoding function $E_a$ for a and the decoding function $D_b$ for b is given by:

$E_a(y) = D_b(y) \equiv y^{Xa} \pmod{p}.$

The decoding function $D_a$ for a and the encoding function $E_b$ for b is given by:

$D_a(y) = E_b(y) \equiv y^{Xb} \pmod{p}.$

## Claims

1. method for the modular reduction of a not more than 2n-digit number x to obtain a not more than n-digit remainder r in accordance with the formula $r \equiv x \pmod p$, where p is the chosen modulus, for the purpose of cryptographic calculations which are based on the general discrete logarithm problem in a number system with base b, characterised in that p is a n-digit number for which it holds true that $p = b^n - a$, where $0 < a < b$.

2. Method according to claim 1, characterized in that the 2n-digit number is always split into two n-digit numbers $x_H$ and $x_L$ in accordance with the formula $x = (x_H * b^n) + x_L$, where

$$x_H = \sum_{i=0}^{n-1} x_{i+n} \cdot b^i \text{ and } x_L = \sum_{i=0}^{n-1} x_i \cdot b^i, \text{ where } 0 \le x_i,\ x_{i+n} < b;$$

in that the remainder r is determined by first calculating:

$r_{temp} = x_L + a * x_H;$

in that it is then determined whether the following is fulfilled:

$r_{temp} < b^n;$

in that, if this is the case, it holds true that:

$r = r_{temp};$

in that, if this is not the case, an integer c is calculated which is determined by the value of $r_{temp}/b^n$ rounded downwards to the nearest integer and $r_{temp}$ is calculated in accordance with:

$r_{temp} = r_{temp} - c * b^n$

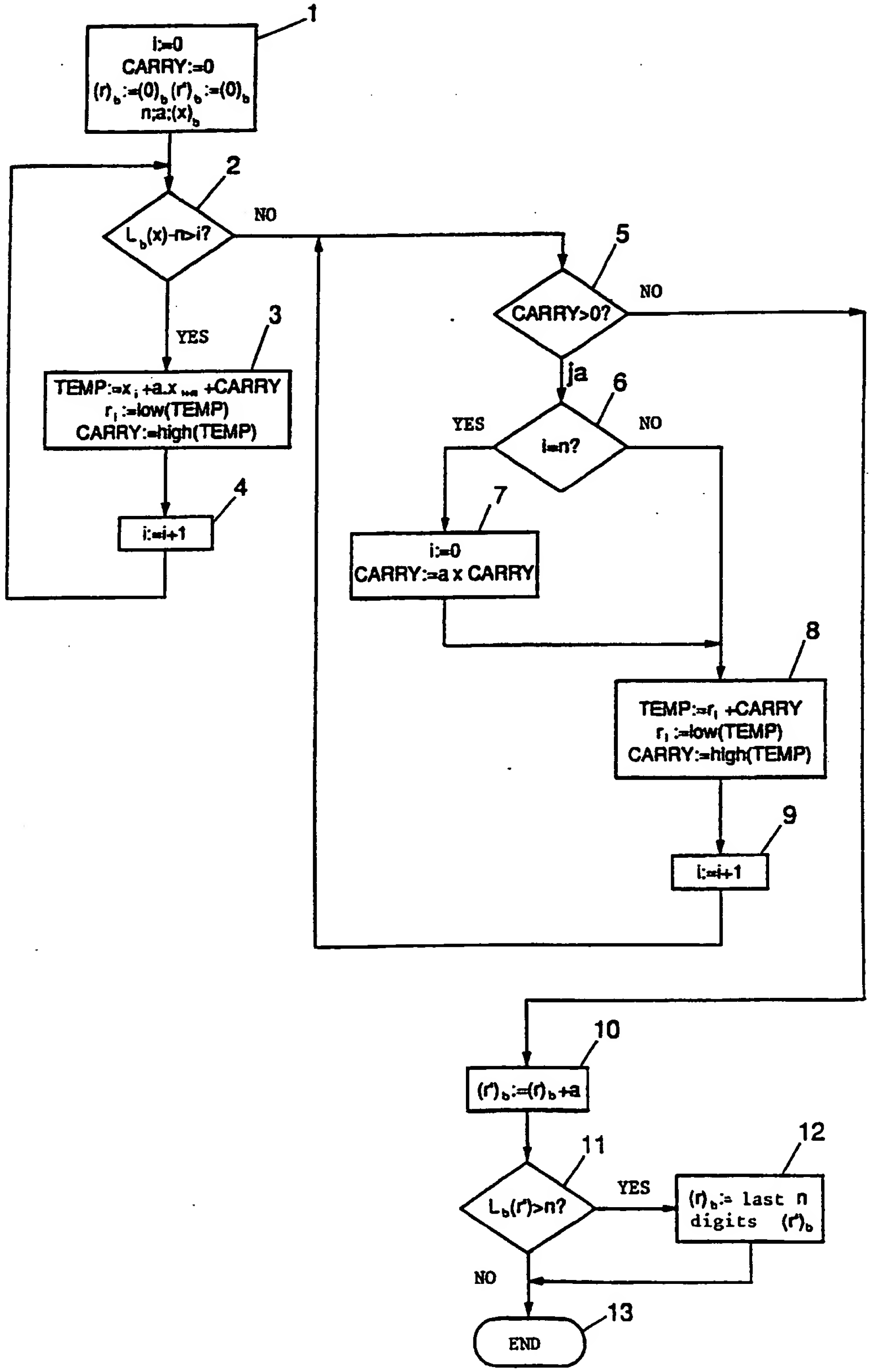
and in that this step is repeated until the following is fulfilled:

$r_{temp} < b^n$,

whereafter it holds true that:

$r = r_{temp}$.

3. Method according to claim 2, characterised in that $r' = r + a$ is calculated, after which the true remainder is equal to the last n digits of r' if $r' > b^n$ and is equal to r if $r' < b^n$.


1
i:=0
CARRY:=0
$(r)_b:=(0)_b$ $(r')_b:=(0)_b$
$n;a;(x)_b$
2
$L_b(x)$-n>i?
NO
YES
3
TEMP:=$x_i$ +a.$x_{i+n}$ +CARRY
$r_i$ :=low(TEMP)
CARRY:=high(TEMP)
4
i:=i+1
5
CARRY>0?
NO
ja
6
i=n?
YES
NO
7
i:=0
CARRY:=a x CARRY
8
TEMP:=$r_i$ +CARRY
$r_i$ :=low(TEMP)
CARRY:=high(TEMP)
9
i:=i+1
10
$(r')_b:=(r)_b$+a
11
$L_b(r')$>n?
YES
NO
12
$(r)_b$:= last n digits $(r')_b$
13
END

European
Patent Office

# EUROPEAN SEARCH REPORT

Application Number

EP 91 20 0523

## DOCUMENTS CONSIDERED TO BE RELEVANT

| Category | Citation of document with indication, where appropriate, of relevant passages | Relevant to claim | CLASSIFICATION OF THE APPLICATION (Int. Cl.5) |
|---|---|---|---|
| X | EP-A-0 350 278 (ATHE HATTFIELD POLYTECHNIC)<br>* Whole document *<br>- - - | 1-3 | G 09 C 1/00<br>G 09 C 5/00<br>H 04 L 9/30 |
| D,A | ELECTRONICS LETTERS, vol. 21, no. 17, 15th August 1985, page 761, London, GB; S.B. MOHAN et al.: "Fast algorithms for implementing RSA public key cryptosystem"<br>* The whole of page 761 *<br>- - - | 1-3 | |
| A | US-A-4 424 414 (HELLMAN et al.)<br>* Column 2, line 36 - column 8, line 8; claims *<br>- - - | 1 | |
| P | PATENT ABSTRACTS OF JAPAN, vol. 14, no. 340 (P-1080)[4283]. 23rd July 1990;<br>& JP-A-2 116 884 (TOSHIBA CORP.) 01-05-1990<br>* Whole document *<br>- - - | 1 | |
| P | EP-A-0 381 523 (TOSHIBA CORP.)<br>* Claims *<br>- - - - - | 1 | |

| TECHNICAL FIELDS SEARCHED (Int. Cl.5) |
|---|
| C 09 C<br>H 04 L |

The present search report has been drawn up for all claims

| Place of search | Date of completion of search | Examiner |
|---|---|---|
| The Hague | 01 July 91 | GORUN M. |

CATEGORY OF CITED DOCUMENTS

X: particularly relevant if taken alone
Y: particularly relevant if combined with another document of the same catagory
A: technological background
O: non-written disclosure
P: intermediate document
T: theory or principle underlying the invention

E: earlier patent document, but published on, or after the filing date
D: document cited in the application
L: document cited for other reasons

&: member of the same patent family, corresponding document